

**CLAIMS**

What is claimed is:

- 1 1. A computer-implemented method for verifying at runtime an invariant property of  
2 a data structure of a computer program, comprising:  
3 automatically generating a first code segment that verifies a runtime value of the  
4 data structure is consistent with the invariant property in response to an annotation of the  
5 data structure that defines the invariant property of the data structure;  
6 comparing the runtime value of the data structure with the invariant property  
7 during execution of the program via execution of the first code segment; and  
8 performing a programmed action if the runtime value is inconsistent with the  
9 invariant property.
- 1 2. The method of claim 1, wherein the invariant property is a range of data addresses  
2 and further comprising verifying that the runtime value of the data structure is within a  
3 range of data addresses specified in source code of the computer program.
- 1 3. The method of claim 1, wherein the invariant property is a range of data addresses  
2 and further comprising:  
3 automatically generating during compilation a valid data address range including  
4 an upper bound and a lower bound for the range of data addresses, wherein the source  
5 code of the computer program does not include a specification of the upper bound and  
6 lower bound; and  
7 verifying that the runtime value of the data structure is within the valid data  
8 address range.

1        5.        The method of claim 1, wherein the invariant property is a range of instruction  
2        addresses and further comprising:  
3                automatically generating during compilation a valid instruction address range  
4        including an upper bound and a lower bound for the range of addresses, wherein the  
5        source code of the computer program does not include a specification of the upper bound  
6        and lower bound; and  
7                verifying that the runtime value of the data structure is within the valid instruction  
8        address range.

1        6.        The method of claim 1, wherein the invariant property is a range of data values and  
2        further comprising the step of verifying that the runtime value of the data structure is  
3        within the range of data values.

1        7.        The method of claim 1, further comprising communicating the invariant property  
2        from a compiler to a code generator.

11

1 9. The method of claim 8, wherein the invariant property is a range of data addresses  
2 and further comprising verifying that the runtime value of the data structure is within a  
3 range of data addresses specified in source code of the computer program.

1 10. The method of claim 8, wherein the invariant property is a range of data addresses  
2 and further comprising:

3 automatically generating during compilation a valid data address range including  
4 an upper bound and a lower bound for the range of data addresses, wherein the source  
5 code of the computer program does not include a specification of the upper bound and  
6 lower bound; and

7 verifying that the runtime value of the data structure is within the valid data  
8 address range.

1 11. The method of claim 8, wherein the invariant property is a range of instruction  
2 addresses and further comprising verifying that the runtime value of the data structure is  
3 within the range of instruction addresses specified in source code of the computer  
4 program.

1 12. The method of claim 8, wherein the invariant property is a range of instruction  
2 addresses and further comprising:

3 automatically generating during compilation a valid instruction address range  
4 including an upper bound and a lower bound for the range of addresses, wherein the

5 source code of the computer program does not include a specification of the upper bound  
6 and lower bound; and  
7 verifying that the runtime value of the data structure is within the valid instruction  
8 address range.

1 13. The method of claim 8, wherein the invariant property is a range of data values and  
2 further comprising the step of verifying that the runtime value of the data structure is  
3 within the range of data values.

1 14. The method of claim 8, further comprising storing in the symbol table one or more  
2 code addresses associated with one or more updates to the data structure.

1 15. An apparatus for verifying at runtime an invariant property of a data structure of a  
2 computer program, comprising:  
3 means for automatically generating a first code segment that verifies a runtime  
4 value of the data structure is consistent with the invariant property in response to an  
5 annotation of the data structure that defines the invariant property of the data structure;  
6 means for comparing the runtime value of the data structure with the invariant  
7 property during execution of the program via execution of the first code segment; and  
8 means for performing a programmed action if the runtime value is inconsistent  
9 with the invariant property.